

Komplisert

Jeg ville brukt react med next.js med Frontend som vil være React med Next.js for rask rendering og ruting og enkle debugging. Backend ville vært supabase for autorisering lagring og infrolagring i sql tasbes der. Jeg ville også lagt inn interctive spill kart med leaf.js men pins. Jeg ville ha bildehosting på cloudwhery pga rask rendering og hvor lett det er å legge i supabase da.

Auth

Sikker Autentisering (Auth)nå ville jeg brukt SHA-256 hashing for sikker sammenligning:

hashPassword(password): En async funksjon som bruker crypto.subtle.digest for å generere en unik hash-streng.

authenticate(user, pass): Sammenligner input-hash med lagret hash fra databasen før det kommer tilgang

Databehandling

fetchGameData(): Henter personlige notater og samleobjekter fra Supabase-tabeller med realtime on

RenderList(): Bruker React-komponenter og .map() (i stedet for en standard for-of løkke) for å da lagde dynamiske liste-elementer () som blir basert på data fra API

Interacitve kart

Jeg ville også istenfor bare å ha bilde og tekst på hvor alt ligger ville jeg laget interavrice kart med en, integrasjon av Leaflet.js

Leaf.js gjør det mulig å plotte inn egne markører på kartet med forklaringer og bilder. Bildene hentes via Cloudinary-URL-er lagret i databasen.

Farger

Jeg ville lagt mange av fargene i global.css som gjør det raskere med å kunne hente det opp litt lettere med dealts så ikke må legge farge på alt med

Element	Fargekode	Beskrivelse
Bakgrunn	#0F1010	Nesten svart

Kort/Paneler	#1D1E20	Kullsvart
Tekst/Ikonter	#ADA6A7	Lys grå for god kontrast
Detaljer	#5B5A58	Dempet grå for skillelinjer

Layout

Nå ville jeg også brukt display none men sleve auth ville skjed i en route.js i app/api/auth/route.js med da SHA-256 så er det api nøkkelen mye mer sikker. Jeg ville spart tid med å lage en egen side som var login.jsx som linker til route.js med lite design så når route.js bekrefter så viderefører den.

Funksjoner som ville blitt brukt

Kategori	Funksjonsnavn	Type	Formål	Teknologi
Autentisering	hashPassword(password)	Async	Genererer en unik hash string	SHA-256 (crypto.subtle)
Autentisering	authenticate(user, pass)	Async	Sammenligner input-hash med lagret hash i supabase	Database-verifikasi
Datahåndtering	fetchGameData()	Async	Henter fra supabase de fordig data til spille	Supabase Realtime API
Datahåndtering	RenderList()	Funksjon	Oppretter dynamiske -	React, Array.map()
Kartintegrasjon	initializeLeafletMap()	Funksjon	interlacet interaktivt Leaflet.js-kart.	Leaflet.js
Kartintegrasjon	plotMapMarkers(data)	Funksjon	Setter pins på kart	Leaflet.js Markers

Kartintegrasjon	<code>fetchCloudinaryImages()</code>	Async	Henter bilde-URL-er lagret i skyen dette kan unngås med å legge til I next.conf filen	Cloudinary API
Ruter / API	<code>handleAuthRoute(request)</code>	Async	autentisering via API-rute med hashing.	Next.js API Routes
Komponenter	<code>LoginComponent()</code>	React	Renderer innloggingsgrense	JSX
Komponenter	<code>GameDetailComponent()</code>	React	Viser spilldetaljer og progresjon for ting.	JSX / Dynamisk ruting
Komponenter	<code>PinComponent()</code>	React	React-DOM komponent for å see av pins.	react-dom, Leaflet.js
Verktøy	<code>applyGlobalStyles()</code>	Funksjon	Bruker globale CSS-stiler fra global.css.	Global CSS
Database	<code>setupSupabaseTables()</code>	Async	Setter opp og integrerer tabellr	Supabase
Database	<code>getUserData(userId)</code>	Async	fetch brukerdata fra databasen.	Supabase

Tidsplannlegging

1-2 time ville satt opp react med next.js og satt opp supabase med riktige talbes og interfrett riktig. Ville også skrevet entyrpen logiken og satt opp fil strukturen. Jeg ville goså etter dette satt opp route.js med pins fra leath.js som krever mange colloms med koridanter og linket til riktig table så dette ville tatt tid.

2-4 front end jeg ville skrevet global.css med også stukteren på login.jsx og skrevet skrevet det lille front ned på ./app/games/[id] etter dette ville nettsiden sett bra ut

5 time

Skrevet ferdig intgresingen av leaf.js sine pins såkl vises med react-dom pin komponenter. Og siden nettsiden bare skal skrives er det ikke regnet tid på å plassere pins.

6 time

En time er satt av til ren debugging ikke noe annet så den foreldes gjennom alt.